

Herramientas de desarrollo con soporte colaborativo en Ingeniería de Software

(Development tools to provide collaboration in Software Engineering)

Julio C. Caiza¹, Danny S. Guamán¹, Gabriel R. López¹

Resumen:

El desarrollo de software es una actividad inherentemente colaborativa. Usualmente los equipos de desarrollo tienen colaboradores localizados en diferentes lugares, hablando diferentes lenguajes e incluso con diferentes culturas. Es necesario conocer sobre aquellas herramientas que pueden ayudar a los desarrolladores en las diferentes etapas del proceso de creación de software, y especialmente en la etapa de codificación. Este artículo muestra los nuevos enfoques que tiene el desarrollo de software colaborativo, su relevancia, y revisa un conjunto de herramientas que incluyen, aquellas creadas como extensiones y aquellas que se usan sobre la web. Se presenta sus características principales y una comparación, ante posibles nuevas implementaciones o para decidir cuál utilizar.

Palabras clave: IDE basados en Web; Colaboración; Conocimiento Compartido; Ingeniería de Software.

Abstract:

Software development is an activity, inherently collaborative; usually development teams have members located in different places, talking different languages and with another culture. Therefore, knowledge about helping tools is necessary. These tools could be helpful in the whole process of software development, and mainly in the coding stage. This paper shows new perspectives in collaborative software development, its relevance; then, shows a set of important tools. They include plugins and web-based tools. A set of important features and a comparison are shown. This information would be helpful if you are thinking about build or use one.

Keywords: Web-based IDE; Collaboration; Awareness Knowledge; Software Engineering.

1. Introducción

Actualmente los proyectos de software, especialmente los de gran tamaño, son inherentemente colaborativos (Whitehead, 2007) esto se debe precisamente a características como la dimensión del proyecto, los tiempos de entrega, e inclusive porque el trabajar en grupo puede ayudar a sobrellevar limitaciones propias de los humanos (Ej. Localización de errores). Con la ayuda del progreso de las comunicaciones e internet, barreras geográficas se han eliminado y ahora los proyectos incluyen personas distribuidas a nivel mundial, de diferentes compañías, idiomas, culturas, etc.

¹ Escuela Politécnica Nacional, Quito – Ecuador ({julio.caiza, danny.guaman, gabriel.lopez} @epn.edu.ec)

El tener varios colaboradores, implica que se tiene varios puntos de vista a medida que se avanza en las etapas del desarrollo de software, haciéndose necesario el llegar a converger en la realización de los diferentes artefactos dentro de cada una de ellas (Maalej & Happel, 2008).

Considerando la importancia de la colaboración, en (Whitehead, 2007) se realizó un importante estudio acerca de la colaboración en el desarrollo de software, realizando una revisión de las herramientas existentes para proveer colaboración en cada una de las etapas. Sin embargo, la evolución de las tecnologías de comunicación y para la web ha hecho que nuevas herramientas y conceptos vayan surgiendo.

La Web 2.0 fundamentada en tecnologías como JavaScript y AJAX ha hecho posible que las aplicaciones web vayan ganando interactividad casi hasta parecerse a aplicaciones de escritorio, por lo que la idea de un IDE (Entorno de Desarrollo Integrado) sobre la web se ha convertido en algo totalmente real. Recursos de la Web 2.0 como redes sociales, blogs, wikis, microblogging, y tagging han sido considerados para mejorar el aspecto colaborativo e informativo en el desarrollo de software.

Considerando el ciclo de vida de desarrollo, la etapa que ha recibido más avance en cuanto a manejar colaboración, es la etapa de codificación y pruebas. Por ende, el enfoque de este artículo es proveer de un estado actual en cuanto a IDEs existentes basadas en la web y que proveen de soporte colaborativo, indicando las tecnologías que han ayudado a la construcción de las mismas.

El resto de este documento se compone de una sección en la que se hacen explícitos los nuevos enfoques que hay en el mundo del desarrollo de software y de la colaboración; en la siguiente sección se revisan extensiones (plugins) para proveer colaboración; a continuación se describe un conjunto de herramientas web completas con soporte a la colaboración; luego se realiza una comparación entre dichas herramientas; la relevancia del uso de las herramientas colaborativas; se establecen las conclusiones; y se finaliza con la sección de trabajo futuro.

2. Nuevos enfoques de colaboración

Los medios sociales se caracterizan por compartir información y se están usando masivamente. Ha servido también para que los investigadores apliquen estas tecnologías en el desarrollo de software, específicamente para dar a conocer sucesos hacia otros usuarios, bien sea dentro de un grupo o fuera de él. En (Storey, Treude, van Deursen, & Cheng, 2010), se realiza un estudio sobre como los medios sociales pueden influir sobre el desarrollo de software, principalmente basados en el hecho de que estas herramientas se caracterizan por tener una arquitectura de participación.

Otro hecho importante es la consideración de aquellos desarrolladores conocidos como End-User; y que son los mismos medios sociales los que ayudan a mejorar su conocimiento, a ordenar su aprendizaje e inclusive a fomentar un aprendizaje en grupo.

2.1. Medios Sociales

Dentro de las herramientas de escritorio se podría mencionar Eclipse que en conjunto con determinadas extensiones, puede guardar la información de las acciones realizadas por un determinado desarrollador; y mostrar luego esta información con un esquema de microblogging.

En la web, los forges son muy populares para los proyectos de código abierto y cada vez integran más características de medios sociales. Claro ejemplo de ello es GitHub que puede trabajar dentro del ámbito de codificación social (Dabbish, Stuart, Tsay, & Herbsleb, 2012). Básicamente GitHub es una red social de desarrolladores con muchas herramientas que ayudan a mantener el trabajo colaborativo entre desarrolladores. Asimismo, ha surgido (Code Pad, 2015) como una comunidad para compartir porciones de código.

Una evolución teniendo en cuenta lo anterior y que se menciona en (Storey et al., 2010), son los Social Development Environment, que son IDEs sobre la web y que involucran además medios sociales. Aunque este entorno es viable para proveer, se debe tener en cuenta:

- Los beneficios y riesgos antes de su implementación.
- Cómo los medios sociales pueden tomar un rol efectivo para soportar coordinación.
- Los tipos de medios sociales que pueden usar para incrementar el conocimiento en un equipo.
- Los problemas inherentes respecto a la privacidad.
- Cómo administrar las interrupciones que se pueden generar mientras se trabaja.
- Finalmente, si su uso implica una mejora en la calidad del software.

Register for free at <https://www.scipedia.com> to download the version without the watermark

Otro enfoque del uso de medios sociales puede ser el reunir retroalimentación de los usuarios finales, esto ayuda a probar que efectivamente el software está trabajando acorde a lo que se quiere o inclusive para llegar a diseñar un mejor producto que se acople a lo que el usuario necesita. Un estudio de este estilo se realiza en (Bajic & Lyons, 2011).

2.2. Desarrolladores End-User

En (Ko et al., 2011) se define a un programador End-User como el rol que una persona asume cuando ella está creando un programa para conseguir algún objetivo a más de escribir un programa. Por lo general este tipo de desarrolladores pueden ir desde aquellos que crean hojas de cálculo hasta aquellos que crean aplicaciones mashup.

En (Singer & Schneider, 2011), se investiga cómo los medios sociales pueden ser usados para juntar programadores End-User para que colaboren entre ellos para intercambiar conocimiento a través de la web. Existen comunidades para estos programadores, donde se aprecia la evolución de este tipo de programadores hasta ser expertos. Incluso, estos entornos se pueden considerar como medios para propagar métodos y mejores prácticas de programación.

En la web pública, los foros pueden ser viables, pero para trabajar con conexiones entre los miembros, el microblogging sería más adecuado. En este tipo de escenarios se requiere establecer mecanismos que fomenten la colaboración de todos.

3. Colaboración a través de extensiones

Las extensiones han servido para poder proveer a los IDEs de funcionalidad extra, en este caso orientadas hacia fomentar el trabajo colaborativo. A continuación se mencionan dos extensiones que se crearon con enfoques distintos.

En (King & Lyons, 2011) se reporta la creación de un plugin para Eclipse, a través del cual cada vez que se realizan eventos en el IDE, estos actualizan el estado un cliente de mensajería instantánea (Google Talk). El objetivo fue evitar las interrupciones que otros mecanismos, como una llamada o un mensaje instantáneo, provocarían. Esta extensión, basada en polling, se orienta a proveer información respecto de presencia, actividad, y lugar en el cual se encuentran.

En (van Deursen et al., 2010) a la par de crear su IDE Adinda, se menciona la creación de una extensión llamada James para Eclipse. La extensión se usa para enviar información hacia Twitter sobre las actividades que se realizan y además permite el realizar anotaciones a cualquier parte del código. Estas interacciones se van guardando en una traza que puede llevar a generar un esquema de pregunta respuesta en una lista.

4. Herramientas Colaborativas sobre la Web

Actualmente, hay un gran número de investigaciones que se orientan a crear herramientas que permitan el desarrollo colaborativo. En general se distinguen por la cantidad de utilidades implementadas y por el modo de controlar la colaboración de los usuarios. A continuación se presenta un grupo de herramientas y se describen sus principales características.

4.1. Wikigramming

Respecto de los medios sociales, se puede considerar el establecer una herramienta basada en el esquema de trabajo de una wiki, debido a la sencillez de la misma. Aunque la mayoría de investigaciones en desarrollo de software usan las wikis como medio de documentación, por ejemplo AdessoWiki (Lotufo, Machado, Körbes, & Ramos, 2009); existen otras como Wikigramming (Hattori, 2011) que es la combinación de una wiki y un intérprete de lenguaje.

La motivación de Wikigramming incluye, que no se necesite instalación ni registro, que se pueda editar libremente el código, y que se puedan ver los cambios inmediatamente. Las características que la herramienta tuvo finalmente son:

- Cada página es asociada a un módulo del programa.
- Su uso es solo para efectos de entrenamiento en desarrolladores iniciales.

- Usa Gauche (Shiro, 2014), un intérprete de scripts realizados en lenguaje Scheme, por lo que no se necesita compilación.
- La manera para revisar el código es a través de unidades de prueba.

4.2. CloudStudio

En (Nordio, Meyer, & Estler, 2011) presenta este IDE que trabaja sobre la web. CloudStudio permite el trabajo de varios desarrolladores de manera simultánea sobre un mismo proyecto y sin importar la ubicación geográfica. Esto debido a que está alojado en la web y trabaja sobre un repositorio de proyectos compartidos. Además, incluye algunas características de social media.

Las características incluyen:

- Respecto a las actualizaciones, no se necesita hacer un commit explícito sino que esto se realiza en segundo plano ante acciones de edición.
- Permite mantener un seguimiento de cambios del resto de colaboradores y permite compilar administrando los cambios.
- Se puede diferenciar el código editado por cada colaborador.
- Permite observar a los usuarios que están activos e iniciar una sesión de chat o Skype con ellos.
- Usa lenguaje Eiffel.
- Incluye una herramienta para hacer pruebas sobre el código.

4.3. CoRED

Register for free at <https://www.scipedia.com> to download the version without the watermark

En (Lautamäki et al., 2012) se presenta las características de CoRED, es un editor de código en tiempo real y colaborativo basado en browser para aplicaciones Java, y acorde a sus autores aún no tiene las suficientes utilidades para ser considerado un IDE. Su arquitectura es totalmente modular, lo que implica que puede ser modificado y extendido de ser necesario.

Las características que presenta son:

- Chequeo de errores.
- Generación de código automático.
- Se pueden colocar notas sobre secciones del código a manera de hilos de comentarios.
- No necesita hacer commits para la sincronización.
- Permite bloqueos sobre partes del código.
- Se basa en Java como lenguaje, Vaadin (Grönroos, 2011) como framework, y Ace (Ace Group, 2015) como frontend.

Vaadin confía en las facilidades de GWT que es un sistema de desarrollo para escribir aplicaciones AJAX en Java y luego compilar el código fuente a código JavaScript, el que corre en

todos los navegadores. Ace es un editor de código, open source, escrito en JavaScript y que puede ser fácilmente embebido en una página web y que soporta varios lenguajes. Al usar Ace, se podrían usar otros lenguajes de desarrollo como JavaScript, HTML, XML, PHP, C++. Al basarse en Java, usa el JDK para compilar y mostrar errores.

4.4. Collabode

En (Goldman, Little, & Miller, 2011) se realiza una descripción de las características de este IDE además de proponer 3 escenarios en los que realizaron las pruebas de la misma. Las características son:

- Usa Eclipse en el lado del servidor, lo que le permite proveer los servicios de compilación continua de errores y warnings, formateo de código, refactoring y ejecución.
- Permite la compilación de cualquier proyecto de Java Eclipse de escritorio.
- El browser solo muestra la salida que procesa el servidor y por lo tanto no se puede ver programas Java con interfaz gráfica.
- Usa EtherPad para soportar colaboración entre usuarios simultáneos.
- Soporta cambios concurrentes.
- Implementa permisos de escritura y lectura sobre proyectos, paquetes, clases y métodos, para controlar la cantidad de información entre los colaboradores.
- Soporta la realización de pruebas. Así un desarrollador puede estar codificando mientras otro puede estar realizando un conjunto de pruebas.
- Provee de una vista para dispositivos móviles.

Register for free at <https://www.scipedia.com> to download the version without the watermark

Su trabajo continúa para dar soporte a otros lenguajes y para soportar el intercambio de información entre colaboradores.

4.5. Adinda

Es un IDE basado que trabaja sobre la web, y se basa en un cliente conectado a una serie de servicios (van Deursen et al., 2010). Las características, que se van añadiendo a medida que surgen nuevas hipótesis, son:

- Tiene servicios para editar, compilar y realizar pruebas.
- Su enfoque colaborativo incluye administración de tareas, edición en grupo, etiquetado, social networking, e incluso data mining.
- Los servicios centralizados recogen todo tipo de información concerniente a las actividades de todo el equipo de desarrollo y lo usa para asistir a desarrolladores individuales.
- Se pueden crear workspaces, proyectos y archivos de clase.
- Edición con colores y con autocompletado.
- En su cliente se pueden hacer programas Java, y el servidor usa Eclipse.

Se basaron en WWWWorkspace (Ryan) y usa AJAX para su interactividad. La filosofía es reutilizar al máximo los componentes.

Los autores reportan consideraciones e implicaciones que van surgiendo, estas tienen que ver con los métodos de versionamiento, estrategias de etiquetado y microblogging para trazabilidad, la participación de los desarrolladores, como identificar la información y el nivel de detalle que hay que guardar; como discernir entre las actividades que en realidad aportan información e inclusive si se puede llegar a establecer recomendaciones.

4.6. Coderun Studio

Acorde a (CodeRun), es un IDE basado en browser. Sus principales características son:

- Soporta la creación de proyectos en varios lenguajes como C#.NET (3.5), PHP (5.1), JavaScript, HTML y CSS.
- Soporte de base de datos SQL Server 2005 y Amazon SimpleDB.
- Creación con el uso de templates.
- En la edición se tiene coloreo de sintaxis y autocompletado de código.
- Para aplicaciones .NET permite compilación y depuración.
- Permite despliegue del proyecto para todos los lenguajes.
- Compartición de código a través de enlaces. A través de este modo se publica un enlace al cual puede acceder un colaborador para editar el código.
- Garantiza la seguridad y privacidad de tu ambiente de desarrollo.

Register for free at <https://www.scipedia.com> to download the version without the watermark

4.7. Cloud9 IDE

En (Daniels, 2015) se define como un IDE basado en browser y que vive en la nube, permitiendo correr, depurar y desplegar aplicaciones desde cualquier lugar y a cualquier hora.

Las características que se señalan son:

- Soporte para los lenguajes JavaScript, HTML, CSS, Coffeescript, Ruby y PHP.
- Su editor tiene soporte de sintaxis y manejo con colores del código.
- Se puede añadir colaboradores para trabajar simultáneamente sobre el mismo proyecto, y maneja permisos sobre los mismos.
- Se puede ejecutar aplicaciones NodeJS dentro de cloud9 de manera fácil.
- Provee de análisis de código en tiempo real.
- Tiene una completa integración con GitHub, provee así de control de versiones, y la posibilidad de compartirlo online.
- Las organizaciones pueden administrar los proyectos asignando o removiendo colaboradores.

- Provee un servicio de chat integrado para la comunicación con los colaboradores.

5. Comparación

En la *Tabla 1* se muestra la comparación de las principales características de las herramientas de desarrollo con soporte colaborativo previamente mencionadas. Durante la búsqueda de información, se han establecido un conjunto de criterios que pueden resultar relevantes al momento de seleccionar una determinada herramienta:

- **Lenguajes soportados.** - Es una característica esencial para que un interesado decida usar una determinada herramienta. La experticia o el interés de un desarrollador en un determinado lenguaje se puede convertir en un requerimiento no funcional muy importante al momento de elegir una herramienta de apoyo.
- **Etapas cubiertas.** - Se refiere a las etapas del proceso de desarrollo de software que han sido cubiertas por la herramienta, mientras más etapas, se vuelve una herramienta más robusta y útil. Hay que tener en cuenta que en fases iniciales del aprendizaje de programación, lo que se pretende dar soporte es la fase de codificación; y en etapas más formales, se da mucha importancia a las etapas de análisis, diseño y pruebas.
- **Sincronización de información.** - Su importancia radica debido al interés de facilitar los procesos al desarrollador. La facilidad o complejidad de la sincronización pueden llevar a que se decida usar o no una herramienta. Además, asociado a esto pueden darse problemas de sincronización en la información y datos que comparten los desarrolladores.
- **Los módulos base** se refieren a los componentes utilizados para construir la herramienta, esta información puede ser muy útil para proyectos que deseen construir herramientas análogas o que busquen por componentes orientados a una actividad en particular. Asimismo, para evitar tecnologías obsoletas o no compatibles con una eventual plataforma existente.
- La comunicación entre colaboradores se vuelve esencial como forma explícita de compartir información con otros desarrolladores. Se vuelve muy importante en casos de conflictos o por dudas ante determinada sección de desarrollo.

Tabla 1. Comparación de las herramientas.

Nombre	Lenguajes Soportados	Etapas Cubiertas	Sincronización Información	Módulos Base	Comunicación Colaboradores
Wikigramming	Scheme	Edición, realización de pruebas.	Similar a Wiki, no simultánea.	Gauche	
CloudStudio	Eiffel	Edición, compilación, depuración, realización de pruebas.	Sincronización en tiempo real.		Chat.
CoRED	Java	Edición, compilación, depuración.	Sincronización en tiempo real.	JDK. Vaadin. Ace.	Etiquetas sobre código.
Collabode	Java	Edición, compilación,	Sincronización en tiempo	Eclipse.	

		depuración, ejecución, realización de pruebas.	real.	Etherpad.	
Adinda	Java	Edición, compilación, realización de pruebas.	Sincronización no en tiempo real, compartición de proyectos a través de enlaces.	Eclipse. WWWorkspace.	Etiquetado.
Coderun Studio	C#/.NET, PHP JavaScript, HTML, CSS	Compilación y depuración para .NET. Despliegue para lenguajes web.	Compartición no en tiempo real, a través de enlaces.		
Cloud9 IDE	JavaScript, HTML, CSS, Coffeescript. Ruby, PHP.	Depuración y despliegue.	Permite trabajo simultáneo sobre proyectos a través de versionamiento.	Usa la tecnología de GitHub	Chat.

Se distinguen dos grupos de herramientas, aquellas que proveen soporte para lenguajes de escritorio compilados y aquellos que proveen soporte para lenguajes de desarrollo web.

Ambos grupos de herramientas trabajan con una arquitectura cliente – servidor. El cliente adquiere su interactividad al usar JavaScript. Respecto al servidor, para el primer grupo, se tiene un compilador que permite detectar los errores de edición y de compilación, los resultados son enviados al cliente a través de AJAX; en el segundo grupo de herramientas, se tiene un módulo que ayuda al despliegue de la aplicación en la web.

El uso de módulos base para construir los IDEs con soporte colaborativo es común, y algunas aplicaciones han logrado obtener un esquema tan modular que inclusive se podría rediseñarlo para soporte para más lenguajes.

Además, el modo de sincronización de datos puede ser realizado en tiempo real gracias a la ayuda de AJAX, o a través de versionamiento usando herramientas como GitHub. Para escenarios donde se tiene una aplicación grande y se requieran de varios colaboradores, la segunda opción podría ser la más acertada.

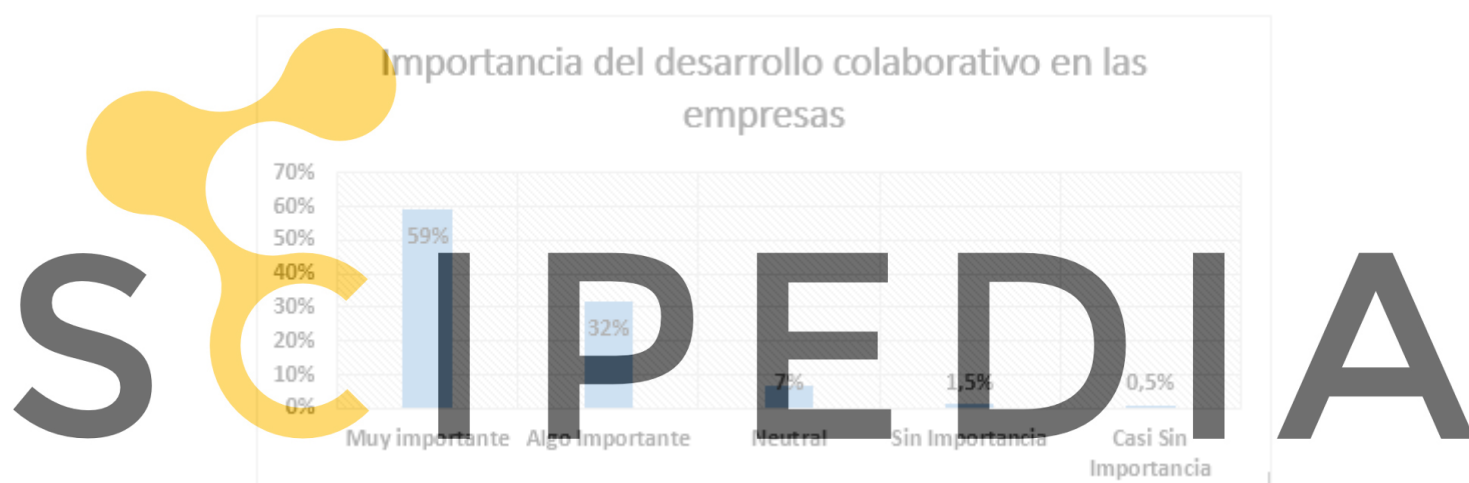
6. Relevancia

La distancia agrava la posibilidad de realizar una coordinación y control efectivos durante un proceso de desarrollo de software debido a que la comunicación se podría ver afectada directa o indirectamente (Carmel, Agarwal 2001). De ahí la relevancia y la diferenciación del uso de herramientas de desarrollo con soporte colaborativo en Ingeniería de Software.

Los ambientes de desarrollo colaborativo tienen como objetivo el aportar en todas las fases de desarrollo de software, es así que empresas que requieran una estrategia confiable de gestión del ciclo de vida del producto asegurando eficiencia, consistencia, y seguridad de la información se sugiere su inversión para un desarrollo colaborativo. Su importancia resalta desde productos iniciales de colaboración como para gestión de versionamiento o gestión de requerimientos, que luego de su implementación tienden a mejorar el desarrollo de software en la empresas (Lanubile, Filippo, et al 2010). En el reporte realizado por (Linux Foundation, 2014) se presentan las

tendencias en el desarrollo colaborativo en las empresas. Dichas empresas, principalmente de tecnología, se enfrentan a una presión considerable para innovar rápidamente y reducir costos. Como consecuencia de ello, se consolida cada vez más un modelo de negocio en donde las empresas, inclusive las competidoras, trabajan de manera colaborativa para compartir recursos de desarrollo y construir bases de código común. Posteriormente, cada uno diferencia sus propios productos y servicios.

En el mismo reporte se puede apreciar que de las 686 respuestas a una encuesta realizada a desarrolladores de software y administradores de negocios, en las que se incluyen a empresas como Cisco, Fujitsu, HP, IBM, Intel, Google, NEC, Oracle, Qualcomm y Samsung, un 91% reconoce la importancia del desarrollo colaborativo en su empresa, tal como se observa en la Figura 1.



Register for free at <https://www.scipedia.com> to download the version without the watermark

Figura 1. Importancia del desarrollo colaborativo en las empresas (Linux Foundation, 2014)

Por otro lado hay que mencionar también que la tendencia de Ingeniería de Software Global se basa en el uso de metodologías de desarrollo ágiles en ambientes distribuidos que inherentemente se basan en la colaboración. La comunicación y colaboración entre desarrolladores y clientes, y la retroalimentación temprana son características esenciales en un proceso de desarrollo iterativo e incremental (Turk, France, & Rumpe, 2014). Además, el desarrollo colaborativo ha sido adoptado por sus beneficios para mejorar la comunicación, coordinación e incrementar la calidad y la productividad (Kaur, Sharma 2014). Siendo la base de esta tendencia de ingeniería de software la colaboración, las herramientas para este propósito demuestran su importancia y justificación de inversión para empresas dedicadas al desarrollo de aplicaciones.

Finalmente, al ser las herramientas de colaboración un mediador entre equipos éste ayuda a romper barreras de lenguaje, diferencia cultural, formación, o de falta de trabajo en equipo (Dorairaj, Noble, Malik, 2011), con el objetivo de no frenar el aporte de personal talentoso, mejorar

la transferencia de conocimiento, reducción de tiempo de desarrollo, mejorar la calidad del software, y dar una respuesta rápida a innovación y cambios en el negocio (Hossain, Babar, Paik 2009). Hay que mencionar también que el desarrollo de aplicaciones Open Source y el desarrollo colaborativo van de la mano. La comunidad del kernel de Linux fue pionero en este enfoque de desarrollo de software, ya que involucra múltiples individuos y empresas, en muchos casos, que compiten en el mismo sector, y en donde el código base es abierto y de una inversión compartida. (Linux Foundation, 2014). Su éxito ha ayudado a inspirar la difusión de métodos de colaboración en otras industrias y tecnologías.

7. Conclusiones

La colaboración en ingeniería de software se ha orientado especialmente a brindar la posibilidad de desarrollar usando herramientas que trabajan sobre la web, y de implementar elementos de social media dentro de las herramientas.

La colaboración puede ser aplicada desde ayudar a los nuevos programadores, incluyendo los End-User; hasta ser aplicada para el desarrollo de aplicaciones grandes para las cuales es necesario disponer de un IDE.

Respecto a la cantidad de herramientas de desarrollo de escritorio y con extensiones, o aquellas que trabajan sobre la web; se observa que el campo de las segundas se ha desarrollado más.

Las herramientas basadas en browser o web se orientan a poder codificar desde cualquier lugar, a cualquier hora, y soportando colaboración.

Una herramienta básica debe contener un editor que soporte colaboración. Aunque actualmente hay verdaderos IDEs que cada vez incorporan mayores funcionalidades.

El diferenciar las herramientas basadas en browser depende de características como los lenguajes que soportan, el modo de sincronización de los datos entre varios colaboradores, el manejo de versiones, y el diseño de su arquitectura.

Se pueden usar los modos de trabajo de las tecnologías de social media para acoplarlos a nuevas maneras de codificación, como por ejemplo el uso de etiquetas y actualizaciones automáticas que se pueden integrar dentro del mismo IDE.

La colaboración entre varios integrantes dentro de un proyecto es soportada cada vez de mejor manera dentro de los IDEs basados en browser, sin embargo, siguen planteándose cuestiones respecto a cuánta información compartir, cómo definirla, y como definir la calidad del mismo.

Muchas herramientas usan una arquitectura modular que usan herramientas producto de investigaciones realizadas previamente, esto implica la ventaja de poder modificarlas acorde a necesidades específicas, pudiendo dar soportes a más lenguajes como a extensiones que permitan colaboración.

Las herramientas que soportan lenguajes compilados distinguen entre el lado del cliente y el lado del servidor, el primero contempla el uso de JavaScript para conseguir la interactividad con el desarrollador, el segundo contempla el uso de un framework que permita compilar, depurar y desplegar la aplicación. Además AJAX tiene un papel muy importante al proveer un medio a través del cual las peticiones y respuestas del servidor se realizan de manera asincrónica permitiendo finalmente conseguir un IDE sobre browsers.

Se ha mostrado la relevancia del campo de estudio a través de reportes que consideran empresas de TI muy conocidas, y que ratifican la importancia de la aplicación de la colaboración en desarrollos globales.

8. Trabajo Futuro

En el contexto de los programadores End-User, hay otra línea de investigación relacionada y que se refiere a la Ingeniería de Software End-User (EUSE - End User Software Engineering), y ésta se orienta al estudio del proceso de desarrollo y de la calidad de software en el desarrollo End-User (Burnett & Myers, 2014). Los autores mencionan una propuesta acerca de romper un enfoque basado en silos, un silo se puede crear por cada etapa dentro de un proceso de desarrollo, e incluso el llegar a usar nuevos modelos de desarrollo que no sigan las etapas dentro del ciclo tradicional. Considerando esta propuesta, ya se puede empezar a pensar en nuevos modelos de desarrollo que aporten a la EUSE y con el soporte adecuado para el desarrollo colaborativo.

Asimismo, considerando que han habido estudios en los cuales se ha usado el enfoque del desarrollo de software colaborativo orientado a mejorar el aprendizaje de ingeniería de software de los estudiantes (Berenson, Williams, Slaten 2005) , y a la productividad y calidad del mismo (Layman, et al. 2005); se puede pensar en que luego de desarrollar modelos diferentes al tradicional, será necesario el determinar nuevas formas de fomentar el aprendizaje de los estudiantes, y de establecer la calidad en los desarrollos.

Considerando las nuevas tendencias que incluyen, la computación en la nube, el grid, el Open data/Linked data, Big Data, el IoT (Internet of Things), la api-ficación, entre otros; ya existen trabajos se están realizando en torno al desarrollo End-User. Así se tienen trabajos como los reportados en (Tetteroo et al., 2015) (Coutaz, Demeure, Caffiau, & Crowley, 2014) que se orientan al desarrollo End-User orientado al IoT o a las Smart Homes. Sin embargo, aún quedan el resto de tendencias en las cuales el desarrollo End-User y el EUSE pueden ser usados. Además, cada una

de las nuevas tendencias puede hacer uso del desarrollo colaborativo para soportar etapas, ayudar a los desarrolladores, enseñar a los desarrolladores novatos o estudiantes, etc.

Bibliografía

- Ace Group. (2015). *The high performance code editor for the web*. Retrieved 04/15, 2015, from <http://ace.c9.io/>
- Bajic, D., & Lyons, K. (2011). Leveraging social media to gather user feedback for software development. *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, Waikiki, Honolulu, HI, USA. pp. 1-6.
- Berenson, S. B., Williams, L., & Slaten, K. M. (2005). Using Pair Programming and Agile Development Methods in a University Software Engineering Course to Develop a Model of Social Interactions. In Crossing Cultures, Changing Lives Conference.
- Burnett, M. M., & Myers, B. A. (2014). Future of End-User Software Engineering: Beyond the Silos. *Proceedings of the on Future of Software Engineering, FOSE 2014* (pp. 201–211). New York, NY, USA: ACM. Retrieved June 6, 2015, from <http://doi.acm.org/10.1145/2593882.2593896>
- Carmel, E., & Agarwal, R. (2001). Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 18(2), 22-29.
- Code Pad. (2015). *Code pad*. Retrieved 04/29, 2015, from <http://codepad.co/>
- CodeRun. *CodeRun*. Retrieved 01/22, 2012, from <http://www.coderun.com/>;
- Coutaz, J., Demeure, A., Caffiau, S., & Crowley, J. L. (2014). Early Lessons from the Development of SPOK, an End-User Development Environment for Smart Homes. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, UbiComp '14 Adjunct* (pp. 895–902). New York, NY, USA: ACM. Retrieved June 6, 2015, from <http://doi.acm.org/10.1145/2638728.2641559>
- Dabbish, L., Stuart, C., Tsay, J., & Herbsleb, J. (2012). Social coding in GitHub: Transparency and collaboration in an open software repository. *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, pp. 1277-1286.
- Daniels, R. (2015). *Cloud 9 your development environment on the cloud*. Retrieved 04/10, 2015, from <https://c9.io/>
- Dorairaj, S., Noble, J., & Malik, P. (2011). Effective communication in distributed Agile software development teams. In *Agile Processes in Software Engineering and Extreme Programming* (pp. 102-116). Springer Berlin Heidelberg.

- Goldman, M., Little, G., & Miller, R. C. (2011). Collabode: Collaborative coding in the browser. *Proceeding of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, pp. 65-68.
- Grönroos, M. (2011). *Book of vaadin* Lulu. com.
- Hattori, T. (2011). Wikigramming: A wiki-based training environment for programming. *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, Waikiki, Honolulu, HI, USA. pp. 7-12.
- Hossain, E., Babar, M. A., & Paik, H. Y. (2009, July). Using scrum in global software development: a systematic literature review. In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on* (pp. 175-184). Ieee.
- Kaur, P., & Sharma, S. (2014). Agile Software Development in Global Software Engineering. *International Journal of Computer Applications*, 97(4), 39-43.
- King, A., & Lyons, K. (2011). Automatic status updates in distributed software development. *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, Waikiki, Honolulu, HI, USA. pp. 19-24.
- Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., et al. (2011). The state of the art in End-User software engineering. *ACM Computing Surveys (CSUR)*, 43(3), 21.
- Lautamäki, J., Nieminen, A., Koskinen, J., Aho, T., Mikkonen, T., & Englund, M. (2012). CoRED—Browser-based collaborative real-time editor for java web applications. *Proceedings of the Fifteenth ACM Conference on Computer Supported Cooperative Work (CSCW 12)*,
- Layman, L., Williams, L., Osborne, J., Berenson, S., Slaten, K., & Vouk, M. (2005, October). How and Why Collaborative Software Development Impacts the Software Engineering Course. In *Frontiers in Education, 2005. FIE'05. Proceedings 35th Annual Conference* (pp. T4C-T4C). IEEE.
- Linux Foundation. (2014). *Collaborative Development Trends Report*. Retrieved from <http://go.linuxfoundation.org//6342/pub-collab-dev-trends-2014/2gw3ly>
- Lotufo, R. A., Machado, R. C., Körbes, A., & Ramos, R. G. (2009). Adessowiki on-line collaborative scientific programming platform. *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, pp. 10.
- Maalej, W., & Happel, H. J. (2008). A lightweight approach for knowledge sharing in distributed software teams. *Practical Aspects of Knowledge Management*, , 14-25.
- Nordio, M., Meyer, B., & Estler, H. -. (2011). Collaborative software development on the web. *CoRR*, abs/1105.0768
- Ryan, W. *WWWWorkspace*. Retrieved 04/15, 2015, from <http://www.willryan.co.uk/WWWWorkspace/>

- Shiro, K. (2014). *Gauche A schem implementation*. Retrieved 04/15, 2015, from <http://practical-scheme.net/gauche/>
- Singer, L., & Schneider, K. (2011). Supporting the cooperation of End-User programmers through social development environments. *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, Waikiki, Honolulu, HI, USA. pp. 13-18.
- Storey, M. A., Treude, C., van Deursen, A., & Cheng, L. T. (2010). The impact of social media on software engineering practices and tools. *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, pp. 359-364.
- Tetteroo, D., Markopoulos, P., Valtolina, S., Paternò, F., Pipek, V., & Burnett, M. (2015). End-User Development in the Internet of Things Era. *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '15 (pp. 2405–2408). New York, NY, USA: ACM. Retrieved June 6, 2015, from <http://doi.acm.org/10.1145/2702613.2702643>
- Turk, D., France, R., & Rumpe, B. (2014). Assumptions underlying agile software development processes.
- van Deursen, A., Mesbah, A., Cornelissen, B., Zaidman, A., Pinzger, M., & Guzzi, A. (2010). Adinda: A knowledgeable, browser-based IDE. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pp. 203-206.
- Whitehead, J. (2007). Collaboration in software engineering: A roadmap. *Future of Software Engineering, 2007. FOSE'07*, pp. 214-225.